MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Language Independent End-to-End Architecture For Joint Language and Speech Recognition

Watanabe, S.; Hori, T.; Hershey, J.R.

TR2017-182 December 2017

Abstract

End-to-end automatic speech recognition (ASR) can significantly reduce the burden of developing ASR systems for new languages, by eliminating the need for linguistic information such as pronunciation dictionaries. This also creates an opportunity, which we fully exploit in this paper, to build a monolithic multilingual ASR system with a language-independent neural network architecture. We present a model that can recognize speech in 10 different languages, by directly performing grapheme (character/chunked-character) based speech recognition. The model is based on our hybrid attention/connectionist temporal classification (CTC) architecture which has previously been shown to achieve the state-of-the-art performance in several ASR benchmarks. Here we augment its set of output symbols to include the union of character sets appearing in all the target languages. These include Roman and Cyrillic Alphabets, Arabic numbers, simplified Chinese, and Japanese Kanji/Hiragana/Katakana characters (5,500 characters in all). This allows training of a single multilingual model, whose parameters are shared across all the languages. The model can jointly identify the language and recognize the speech, automatically formatting the recognized text in the appropriate character set. The experiments, which used speech databases composed of Wall Street Journal (English), Corpus of Spontaneous Japanese, HKUST Mandarin CTS, and Voxforge (German, Spanish, French, Italian, Dutch, Portuguese, Russian), demonstrate comparable/superior performance relative to language-dependent end-to-end ASR systems.

IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)

LANGUAGE INDEPENDENT END-TO-END ARCHITECTURE FOR JOINT LANGUAGE IDENTIFICATION AND SPEECH RECOGNITION

Shinji Watanabe, Takaaki Hori, and John R. Hershey

Mitsubishi Electric Research Laboratories (MERL), Cambridge MA, USA

ABSTRACT

End-to-end automatic speech recognition (ASR) can significantly reduce the burden of developing ASR systems for new languages, by eliminating the need for linguistic information such as pronunciation dictionaries. This also creates an opportunity, which we fully exploit in this paper, to build a monolithic multilingual ASR system with a language-independent neural network architecture. We present a model that can recognize speech in 10 different languages, by directly performing grapheme (character/chunked-character) based speech recognition. The model is based on our hybrid attention/connectionist temporal classification (CTC) architecture which has previously been shown to achieve the state-of-the-art performance in several ASR benchmarks. Here we augment its set of output symbols to include the union of character sets appearing in all the target languages. These include Roman and Cyrillic Alphabets, Arabic numbers, simplified Chinese, and Japanese Kanji/Hiragana/Katakana characters (5,500 characters in all). This allows training of a single multilingual model, whose parameters are shared across all the languages. The model can jointly identify the language and recognize the speech, automatically formatting the recognized text in the appropriate character set. The experiments, which used speech databases composed of Wall Street Journal (English), Corpus of Spontaneous Japanese, HKUST Mandarin CTS, and Voxforge (German, Spanish, French, Italian, Dutch, Portuguese, Russian), demonstrate comparable/superior performance relative to language-dependent end-to-end ASR systems.

Index Terms— End-to-end ASR, multilingual ASR, language-independent architecture, language identification, hybrid attention/CTC

1. INTRODUCTION

End-to-end automatic speech recognition (ASR) has shown the effectiveness recently by greatly reducing ASR building procedure, and by reaching the state-of-the-art performance obtained by conventional hybrid systems [1, 2, 3, 4]. There are two main streams for end-to-end ASR systems by using connectionist temporal classification (CTC) [5, 6, 7] and attention-based encoder-decoder approaches [8, 9, 10, 11]. The unique property of the end-to-end ASR systems is that the network directly converts input speech feature sequences to output label sequences (mainly character or chunked-character in this paper) without through any phonetic/linguistic representation including phonemes or words. Thus, one of the biggest advantages of the end-to-end ASR systems is that it can reduce the efforts of language-dependent processing including the use of pronunciation dictionary and word segmentation, which is a big barrier when we build a conventional ASR system for new languages.

This paper fully exploits the above advantage, and proposes to build a monolithic multilingual ASR system with a language-independent neural network architecture. The model is based on our hybrid attention/CTC architecture [12]. This model compensates too flexible alignment properties in the attention-based method with CTC as a regularization during training and as a score correction during decoding [13]. The model is extended to use a deep convolutional neural network (CNN) followed by bidirectional long short-term memory (BLSTM) in the encoder networks, and recurrent neural network language model (RNN-LM) pretrained with text data in the decoder network. This extended hybrid attention/CTC obtains comparable/superior performance to the state-of-the-art deep neural network (DNN)/hidden Markov model (HMM) baselines including lattice-free maximum mutual information (MMI) training [14] in large vocabulary Japanese/Mandarin speech recognition tasks [3].

Based on this hybrid attention/CTC, we present a model that can recognize speech in 10 different languages composed of Wall Street Journal (English) [15, 16], Corpus of Spontaneous Japanese [17], HKUST Mandarin CTS [18], and Voxforge (German, Spanish, French, Italian, Dutch, Portuguese, Russian) [19]. The model directly performs grapheme (character/chunked characters) based speech recognition conforming to the end-to-end fashion. To make the model language-independent, we augment its set of output symbols to include the union of character sets appearing in all the target languages. These include Roman and Cyrillic Alphabets, Arabic numbers, simplified Chinese, and Japanese Kanji/Hiragana/Katakana characters (5,500 characters in all). This allows training of a single multilingual model, whose parameters are shared across all the languages. In addition to this language-independent architecture, the model also predicts a language ID as well as a text output. Thus, the model can jointly identify the language and recognize the speech, automatically formatting the recognized text in the appropriate character set. The experiments demonstrate comparable/superior performance relative to language-dependent end-to-end ASR systems.

2. RELATED WORK

There have been a lot of prior studies on multilingual/language-independent ASR (see [20] for more details). Initial attempts are based on Gaussian mixture model (GMM)/HMM based acoustic model [21, 22], which are extended with DNNs [23] [24] [25]. DNN-based approaches are also used to produce multilingual bottleneck features [26] [27]. All of these approaches are based on a phoneme representation based on discrete phonetic symbols. To obtain the phoneme transcripts required in the above approaches, we have to prepare hand-crafted pronunciation dictionary for every language. However, our proposed method uses end-to-end systems, and does not require such explicit phoneme representation.

In addition, to build multilingual/language-independent ASR, many systems listed above have language-dependent modules in either or both of acoustic and language models. Therefore, the system

has to know which language is uttered in advance by combining a language identification module [28] [29]. On the other hand, the proposed system is fully a language-independent architecture, and it does not require the language identification module. However, it is beneficial to provide language ID information for the latter application or interfaces, and our proposed system supplementarily predicts a language ID in addition to a target language text.

3. HYBRID ATTENTION/CTC ARCHITECTURE

This section briefly explains our hybrid attention/CTC architecture, which utilizes both benefits of CTC and attention during training and decoding [3].

3.1. Connectionist Temporal Classification (CTC)

CTC [5] is a latent variable model that monotonically maps an input sequence to an output sequence of shorter length. We assume here that the model outputs L-length character sequence $C = \{c_l \in \mathcal{U} | l = 1, \cdots, L\}$ with a set of distinct characters $\mathcal{U}$. CTC introduces framewise character sequence with an additional "blank" symbol $Z = \{z_t \in \mathcal{U} \cup \text{blank} | t = 1, \cdots, T\}$. By using conditional independence assumptions, the posterior distribution $p(C|X)$ is factorized as follows:

$$p(C|X) \approx \underbrace{\sum_{Z} \prod_{t} p(z_t|z_{t-1}, C) p(z_t|X)\, p(C)}_{\triangleq p_{\text{ctc}}(C|X)} \tag{1}$$

As shown in Eq. (1), CTC has three distribution components; framewise posterior distribution $p(z_t|X)$, transition probability $p(z_t|z_{t-1}, C)$, and character-based language model $p(C)$. The CTC objective function $p_{\text{ctc}}(C|X)$, which does not include the language model, is also defined for the use of the later formulation.

This paper uses a deep CNN/BLSTM network to obtain the framewise posterior distribution $p(z_t|X)$ conditioned on all inputs X:

$$p(z_t|X) = \text{Softmax}(\text{Lin}(\mathbf{h}_t)) \tag{2}$$

$$\mathbf{h}_t = \text{BLSTM}(\text{CNN}(X)). \tag{3}$$

Softmax$(\cdot)$ is a softmax activation function, and Lin$(\cdot)$ is a linear layer to convert hidden vector $\mathbf{h}_t$ to a $(|\mathcal{U}| + 1)$ dimensional vector ($+1$ means a blank symbol introduced in CTC). CNN$(\cdot)$ is a CNN layer followed by a BLSTM layer BLSTM$(\cdot)$.

The use of a deep CNN architecture is motivated by the prior studies [30, 31]. We use the initial 6 layers of the VGG net architecture [32]:

Convolution2D(# in $= 3$, # out $= 64$, filter $= 3 \times 3$)
Convolution2D(# in $= 64$, # out $= 64$, filter $= 3 \times 3$)
Maxpool2D(patch $= 3 \times 3$, stride $= 2 \times 2$)
Convolution2D(# in $= 64$, # out $= 128$, filter $= 3 \times 3$)
Convolution2D(# in $= 128$, # out $= 128$, filter $= 3 \times 3$)
Maxpool2D(patch $= 3 \times 3$, stride $= 2 \times 2$)

The initial three input channels are composed of the spectral features, delta, and delta delta features. Input speech feature images are downsampled to $(1/4 \times 1/4)$ images along with the time-frequency axes through the two max-pooling (Maxpool2D) layers.

Although Eq. (1) has to deal with a summation over all possible Z, we can efficiently compute this marginalization by using dynamic programming thanks to the Markov property.

3.2. Attention-based encoder-decoder

Compared with CTC approaches, the attention-based approach does not make any conditional independence assumptions, and directly estimates the posterior $p(C|X)$ based on the chain rule:

$$p(C|X) = \underbrace{\prod_{l} p(c_l|c_1, \cdots, c_{l-1}, X)}_{\triangleq p_{\text{att}}(C|X)}, \tag{4}$$

where $p_{\text{att}}(C|X)$ is an attention-based objective function. $p(c_l|c_1, \cdots, c_{l-1}, X)$ is obtained by

$$p(c_l|c_1, \cdots, c_{l-1}, X) = \text{Decoder}(\mathbf{r}_l, \mathbf{q}_{l-1}, c_{l-1}) \tag{5}$$

$$\mathbf{h}_t = \text{Encoder}(X) \tag{6}$$

$$a_{lt} = \text{Attention}(\{a_{l-1}\}_t, \mathbf{q}_{l-1}, \mathbf{h}_t) \tag{7}$$

$$\mathbf{r}_l = \sum_t a_{lt} \mathbf{h}_t. \tag{8}$$

Eq. (6) converts input feature vectors X into a framewise hidden vector $\mathbf{h}_t$ in an encoder network based on CNN/BLSTM, i.e., Encoder$(X) \triangleq$ BLSTM(CNN(X)), which is the same structure as Eq. (3). Attention$(\cdot)$ in Eq. (7) is based on a content-based attention mechanism with convolutional features, as described in [9]. a_{lt} is an attention weight, and represents a soft alignment of hidden vector $\mathbf{h}_t$ for each output c_l based on the weighted summation of hidden vectors to form character-wise hidden vector $\mathbf{r}_l$ in Eq. (8). A decoder network is another recurrent network conditioned on previous output c_{l-1} and hidden vector $\mathbf{q}_{l-1}$, similar to RNN-LM, in addition to character-wise hidden vector $\mathbf{r}_l$. We use Decoder$(\cdot) \triangleq$ Softmax(Lin(LSTM$(\cdot)$)).

Compared with CTC, attention-based models make predictions conditioned on all the previous predictions, and thus can learn language-model-like output contexts. However, the cost of using an explicit alignment without monotonic constraints means the alignment can become impaired.

3.3. Multi-task learning

In [12], we used the CTC objective function as an auxiliary task to train the attention model encoder within the multi-task learning (MTL) framework. This approach substantially reduces irregular alignments during training and inference, and provides improved performance in several benchmarks. The hybrid attention/CTC shares the same CNN/BLSTM encoder with CTC and attention decoder networks in Eqs. (3) and (6). Unlike the sole attention model, the forward-backward algorithm of CTC can enforce monotonic alignment between speech and label sequences during training. That is, rather than solely depending on the data-driven attention mechanism to estimate the desired alignments in long sequences, the forward-backward algorithm in CTC helps to speed up the process of estimating the desired alignment. The objective to be maximized is a logarithmic linear combination of the CTC and attention objectives, i.e., $p_{\text{ctc}}(C|X)$ in Eq. (1) and $p_{\text{att}}(C|X)$ in Eq. (4):

$$\mathcal{L}_{\text{MTL}} = \lambda \log p_{\text{ctc}}(C|X) + (1 - \lambda) \log p_{\text{att}}(C|X), \tag{9}$$

with a tunable parameter $\lambda : 0 \leq \lambda \leq 1$.

3.4. Joint decoding

In addition to use the CTC objective through multi-task learning, we use the CTC predictions also in the decoding process. The inference

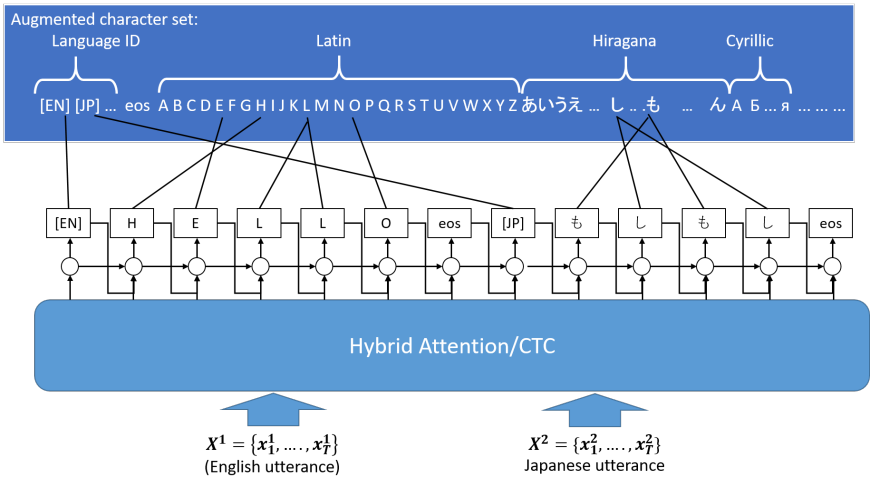

Fig. 1. The proposed language-independent architecture. The system learns to predict the language ID in the beginning of an utterance followed by a text output.

step of attention-based speech recognition is performed by output-label synchronous decoding with a beam search. However, we take the CTC probabilities into account to find a better aligned hypothesis to the input speech, i.e. the decoder finds the most probable character sequence $\hat{C}$ given speech input X, according to

$$\hat{C} = \arg\max_{C \in \mathcal{U}^*} \{\lambda \log p_{\text{ctc}}(C|X) + (1-\lambda) \log p_{\text{att}}(C|X)\}. \quad (10)$$

In the beam search process, the decoder computes a score of each partial hypothesis. With the attention model, the score can be computed recursively as

$$\alpha_{\text{att}}(g_l) = \alpha_{\text{att}}(g_{l-1}) + \log p(c|g_{l-1}, X), \quad (11)$$

where g_l is a partial hypothesis with length l, and c is the last character of g_l, which is appended to g_{l-1}, i.e. $g_l = g_{l-1} \cdot c$. The score for g_l is obtained as the addition of the original score $\alpha(g_{l-1})$ and the conditional log probability given by the attention decoder in (5). During the beam search, the number of partial hypotheses for each length is limited to a predefined number, called a *beam width*, to exclude hypotheses with relatively low scores, which dramatically improves the search efficiency.

3.5. Decoder with RNN-LM

Finally, we combine an RNN-LM network in parallel with the attention decoder. The hybrid attention/CTC and RNN-LM is trained separately, where the RNN-LM is trained with character sequences without word-level knowledge. Although the attention decoder implicitly includes a language model as in Eq. (5), we aim at introducing language model states purely dependent on the output label sequence in the decoder, which potentially brings a complementary effect.

The RNN-LM probabilities are used to predict the output label jointly with the decoder network. The RNN-LM information is combined in the log-probability domain, as follows:

$$\hat{C} = \arg\max_{C \in \mathcal{U}^*} \{\lambda \log p_{\text{ctc}}(C|X) + (1-\lambda) \log p_{\text{att}}(C|X) + \gamma \log p_{\text{rnnlm}}(C)\}, \quad (12)$$

where γ is an additional scaling parameter for RNN-LMs. Although it is possible to apply the RNN-LM as a rescoring step, we combine the RNN-LM network in the end-to-end model because we do not wish to have an additional rescoring step for better latency. Also, we can view this as a single large neural network model, even if parts of it are separately pretrained. Furthermore, [3] also proposes to train the RNN-LM and hybrid attention/CTC jointly, but this paper only uses a pretrained RNN-LM.

4. LANGUAGE-INDEPENDENT ARCHITECTURE

This section explains the proposed monolithic multilingual ASR system with a language-independent neural network architecture, as shown in Figure 1. All the network parameters are shared across languages including output softmax layer, which is represented by the following augmented character set.

4.1. Augmented character set

We augment its set of output symbols to include the union of character sets appearing in all the target languages, i.e.,

$$\mathcal{U} = \mathcal{U}^{\text{EN}} \cup \mathcal{U}^{\text{JP}} \cup \cdots, \quad (13)$$

where $\mathcal{U}^{\text{EN/JP/}\cdots}$ is a character set of a specific language. The advantage of using this augmented character set is to accept any language without language identification modules. The network learns to predict a character sequence in a target language, automatically. However, since we do not explicitly constrain the character set for each language, there is a risk that the language can be switched to the others during an utterance. However, our preliminary experiments show that this language switch was not observed frequently, probably due to the strong context modeling in the decoder network.

Table 1. Multilingual ASR tasks using Wall Street Journal (English) [15, 16], Corpus of Spontaneous Japanese [17], HKUST Mandarin CTS [18], and Voxforge (German, Spanish, French, Italian, Dutch, Portuguese, Russian) [19]. The voxforge data were downloaded at May 2017.

Corpus	Tasks	# utterances	Lengths (h)
WSJ English (**EN**)	Training (WSJ1 SI284)	37,416	80
	Development (Dev93)	503	1.1
	Evaluation (Eval92)	333	0.7
CSJ Japanese (**JP**)	Training (100k)	100,000	147
	Training (Full)	445,068	581
	Evaluation (task 1)	1,288	1.9
	Evaluation (task 2)	1,305	2.0
	Evaluation (task 3)	1,389	1.3
HKUST Mandarin (**CH**)	Training (100k)	100,000	90
	Training (speed perturb.)	580,161	501
	Development	4,000	4.8
	Evaluation	5,413	4.9
Voxforge German (**DE**)	Training	33,272	45.5
	Development	4,033	5.5
	Evaluation	4,017	5.5
Voxforge Spanish (**ES**)	Training	17,954	40.3
	Development	1,583	3.2
	Evaluation	2,663	6.9
Voxforge French (**FR**)	Training	17,882	29.4
	Development	2,386	3.9
	Evaluation	2,218	3.5
Voxforge Italian (**IT**)	Training	8,362	15.7
	Development	1,078	2.0
	Evaluation	1,044	2.0
Voxforge Dutch (**NL**)	Training	6,739	8.3
	Development	868	1.0
	Evaluation	865	1.0
Voxforge Portuguese (**PT**)	Training	2,778	2.9
	Development	352	0.3
	Evaluation	306	0.3
Voxforge Russian (**RU**)	Training	5,009	11.9
	Development	662	1.6
	Evaluation	588	1.2

4.2. Joint language identification and speech recognition

Although our language-independent end-to-end ASR can implicitly predict a target language, it would be useful when we also provide the predicted language ID explicitly for several applications. To do this extension, we introduce an additional variable $k \in \{\text{EN}, \text{JP}, \cdots\}$ indicating a language ID, and deal with the joint distribution of a language ID and text as $p(k, C|X)$ instead of $p(C|X)$ in the attention-based approach. This is formulated by using the probabilistic chain rule, used in Section 3.2, as follows:

$$p(k, C|X) = p(k) \prod_l p(c_l|k, c_1, \cdots, c_{l-1}, X), \qquad (14)$$

This scheme is easily implemented with the current attention-based ASR system by introducing language index in the beginning of the output text in the training data, as follows:

- "こ れ ら の"
  → "[JPN] こ れ ら の"

- "A L S O ␣ V O R W Ä R T S"
  → "[DE] A L S O ␣ V O R W Ä R T S"

Table 2. Experimental configuration

Parameter initialization	uniform dist. [-0.1, 0.1]
# of encoder BLSTM cells	320
# of encoder projection units	320
# of decoder LSTM cells	300
Optimization	AdaDelta
AdaDelta ρ	0.95
AdaDelta ϵ	10^{-8}
AdaDelta ϵ decaying factor	10^{-2}
Gradient norm clip threshold	5
Maximum epoch	15
Threshold to stop iteration	10^{-4}
Location-aware # of conv. filters	10
Location-aware conv. filter widths	100
Hybrid attention/CTC λ	0.5
RNN-LM weight γ	0.1

To do this end, we further augment the character set to include the language id, i.e., $\mathcal{U}^{\text{final}} = \mathcal{U} \cup \{\text{EN}, \text{JP}, \cdots\}$. Note that with the probabilistic chain rule, we could insert a language index into any positions. However, setting a language index in the beginning is straightforward since it first predicts a language ID, and performs ASR conditioned on the predicted ID, which behaves similar to the conventional scheme having a language identification module as prepossessing.

Thus, the augmented feature set and joint language identification and speech recognition enable language-independent multilingual ASR within an end-to-end fashion.

5. EXPERIMENTS

5.1. Setup

This section demonstrates multilingual ASR experiments with our proposed language-independent end-to-end system. Table 1 shows corpora based on WSJ, CSJ, HKUST, and Voxforge. For the Voxforge data, we randomly split them with 80% for a training set, 10% for a development set, and the rest of 10% for an evaluation set[1] Note that the size of corpora for each language is not well balanced (Japanese and augmented Mandarin corpora have more than 500 hours while the Portuguese corpus only has 2.9 hours). We first performed a relatively small-scale experiment based on a subset of database by using 7 larger scale languages (i.e., JP, CH, EN, DE, ES, FR, IT), and used a part of utterances (100K) for Japanese and Mandarin (we call it **7lang**). The total amount of **7lang** training data is 449 hours while that of full training data (**10lang**) is 1327 hours.

Table 2 lists the common experimental hyperparameters among all experiments. When we built language-independent and language-dependent models, we use the exactly same hyperparameters listed in the table, except for the encoder network architecture (number of BLSTM layers and the use of the CNN layer). Each BLSTM in the encoder network has 320 cells. A full connected layer is inserted between BLSTM layers, which linearly transforms two-directional outputs (640 dims.) to 320 dimensional vectors. The decoder network consists of 1 layer LSTM with 300 cells.

To use the same dimensional input features, we used 40-dimensional filterbank features with 3-dimensional pitch features

[1] More specifically, we split the database by making prompts open to each other. Voxforge often uses the same prompts for several utterances, and we have to avoid the same prompts appeared in the training and test data.

Table 3. Character Error Rates (CERs) of language-dependent and language-independent ASR experiments for 7 and 10 multilingual setups.

			Language-dependent 4BLSTM	**7lang** 4BLSTM	**7lang** CNN-7BLSTM	**7lang** CNN-7BLSTM RNN-LM	**10lang** CNN-7BLSTM RNN-LM
HKUST	CH	train_dev	40.1	43.9	40.5	40.2	32.0
		dev	40.4	43.6	40.5	40.0	31.0
WSJ	EN	dev93	9.4	9.6	7.7	7.0	9.7
		eval92	7.4	7.3	5.6	5.1	7.4
CSJ	JP	eval1	13.5	14.3	12.4	11.9	10.2
		eval2	10.8	10.8	9.0	8.5	7.2
		eval3	23.2	24.9	22.0	21.4	8.7
Voxforge	DE	dev	6.6	7.4	5.7	5.4	7.3
		eval	5.2	7.4	5.8	5.5	7.3
	ES	dev	50.9	28.1	31.9	31.5	25.8
		eval	50.8	29.6	34.7	34.4	26.7
	FR	dev	27.7	25.0	22.0	21.0	24.1
		eval	26.5	23.5	21.2	20.3	23.2
	IT	dev	14.3	14.3	11.8	11.1	13.8
		eval	14.3	14.4	12.0	11.2	14.1
	NL	dev	27.0				23.2
		eval	25.5				22.4
	RU	dev	47.8				45.0
		eval	49.4				43.2
	PT	dev	56.9				35.5
		eval	52.2				31.9
Avg.	7 langs		22.7	20.3	18.9	18.3	**16.6**
Avg.	10 langs		27.4				**21.4**

implemented in Kaldi [33] for both 8/16 kHz speech signals[2]. With CNN/BLSTM-based encoder network, we used additional delta and delta-delata features to form 3-channel inputs in the CNN. Our initial experiments only used BLSTM as an encoder network, and in this configuration, we only used the static 43-dimensional feature and subsampled hidden output activations on 1st and 2nd bottom layers (skip every 2nd feature, yielding $4/T$).

The language-dependent multilingual ASR model was trained for each of 10 languages. Similar to the **7lang** setup, we only used subsets for Japanese (150 hours from a subset of CSJ training data) and Mandarin (90 hours from a subset of HKUST CTS training data) corpora. This paper also strictly followed an end-to-end ASR concept, and did not use any pronunciation lexicon, word-based language model, GMM/HMM, or DNN/HMM. Our hybrid attention/CTC architecture was implemented with Chainer [34].

5.2. Results

Table 3 shows the character error rate (CER) of language-dependent and language-independent end-to-end ASR systems with several experimental configurations. The first experiment is to compare the language-dependent and language-independent end-to-end ASR systems with the same network architecture, based on relatively small-scale setup. To do this comparison, we only used 4-layer BLSTM instead of CNN/BLSTM in the encoder network, and also limit the training data with **7lang** for language-independent architecture. Columns "Language-dependent 4BLSTM" and "**7lang** 4BLSTM" corresponds to this comparison. The language-independent ASR system successfully improved the performance in average by 2.4%, mainly improving the performance of Spanish task, which was extremely poor performance on a language-dependent setup. Although the average performance was improved, the performance of many languages were actually degraded probably due to the straightforward mixing of all languages into a sigle network.

The second experiments in the "**7lang** CNN-7BLSTM" and "**7lang** CNN-7BLSTM, RNN-LM" columns enhanced the network architecture by using the CNN followed by the 7-layer BLSTM instead of the 4-layer BLSTM in the encoder network, and also combining RNN-LM, as described in Section 3. We first prepared a language-independent LSTM-based RNN-LM with 800 cell size trained by mixing the transcripts of **7lang** with the same augmented character set as the language-independent end-to-end architecture. These two extensions significantly improved the performance by 2.0% absolutely in average, and also recovered most of degradations observed in the previous experiments. We only observed marginal degradation on the HKUST train_dev and dev, and Voxforge German evaluation set.

Given the success of our very deep encoder network, the final experiment in the column "**10lang** CNN-7BLSTM, RNN-LM" used full training data of 10 language with the same CNN-7BLSTM architecture in the encoder network. Similarly to the previous experiment, we also prepared a language-independent LSTM-based RNN-LM with 800 cell size trained by mixing the transcripts of **10lang**. With the training data extension, we achieved further improvement for the 7 language test sets with 16.6% CER. We also obtained 21.4% CER for the 10 language test set. Although it cannot be directly compared with the average CER of 27.4% obtained by language-dependent systems due to the different network architectures and different amounts of training data, we could still

[2]The features obtained from 8/16 kHz on this setup are not consistent, and we may need to compensate this feature difference, which is one of our future work.

		CH	EN	JP	DE	ES	FR	IT	NL	RU	PT
	train_dev	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CH	dev	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	test_eval92	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
EN	test_dev93	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	eval1_jpn	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	eval2_jpn	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
JP	eval3_jpn	0.0	0.0	99.9	0.0	0.0	0.0	0.1	0.0	0.0	0.0
	et_de	0.0	0.0	0.0	99.7	0.0	0.0	0.0	0.3	0.0	0.0
DE	dt_de	0.0	0.0	0.0	99.7	0.0	0.0	0.0	0.3	0.0	0.0
	dt_es	0.0	0.0	0.0	0.0	67.9	0.0	31.9	0.0	0.0	0.2
ES	et_es	0.0	0.0	0.0	0.1	91.1	0.0	8.4	0.1	0.0	0.2
	dt_fr	0.0	0.0	0.0	0.1	0.0	99.4	0.0	0.2	0.0	0.3
FR	et_fr	0.0	0.0	0.0	0.1	0.0	99.5	0.0	0.1	0.0	0.3
	dt_it	0.0	0.0	0.0	0.0	0.3	0.4	99.1	0.0	0.0	0.3
IT	et_it	0.0	0.0	0.0	0.0	0.4	0.4	98.3	0.2	0.1	0.7
	dt_nl	0.0	0.0	0.0	1.3	0.0	0.1	0.1	97.2	0.0	1.3
NL	et_nl	0.0	0.0	0.0	1.0	0.0	0.2	0.2	97.6	0.0	0.9
	dt_ru	0.2	0.0	0.0	0.0	0.2	0.6	0.5	0.0	97.9	0.8
RU	et_ru	0.0	0.0	0.0	0.2	0.2	0.3	4.3	0.0	94.7	0.3
	dt_pt	0.0	0.0	0.0	0.3	0.3	2.6	1.7	3.4	0.6	91.2
PT	et_pt	0.0	0.3	0.0	0.3	0.0	0.0	3.9	3.6	0.3	91.5

Fig. 2. Language identification (LID) accuracies/error rates (%). The diagonal elements correspond to the LID accuracies while the off-diagonal elements correspond to the LID error rates.

state that our language-independent end-to-end systems achieved reasonable performance with 10 multiple languages. Also, note that among worst three performance languages (Spanish (ES), Russian (RU), and Portuguese (PT)) in the language-dependent condition, ES and PT significantly improved the performance in the language-independent codtion, while RU did not. One reason of this different trend is that ES and PT shared common graphemes (Roman Alphabet) with the other languages, and their sparse data issues would be largely mitigated by the training data of the other languages with our language-independent monolithic architecture. However, since the Cyrillic Alphabet in RU is not appeared in the other languages, the RU task could not obtain the benefit from the training data of the other languages, which would yield the marginal improvement compared with ES and PT.

Finally, Figure 2 shows the language identification (LID) accuracies/error rates (%) for **10lang** CNN-7BLSTM, RNN-LM. The diagonal elements correspond to the LID accuracies while the off-diagonal elements correspond to the LID error rates. We can observe that the language identification is almost perfect except for Spanish, which tended to be mis-recognized as Italian, since Spanish and Italian are linguistically close to each other.

Thus, we confirmed that the proposed architecture can realize language-independent speech recognition with high performance language identification at the same time.

6. SUMMARY

We proposed a language-independent ASR architecture, and shows the effectiveness of the proposed architecture. The architecture does not have language-dependent components, and accepts speech input of any target languages. One of the current issues is that the ASR performance of several languages was degraded mainly due to the unbalanced training data, which is always happened in the field data. Our future work is to mitigate this unbalanced training data issue, and perform language-independent ASR without the degradation of specific languages.

7. REFERENCES

[1] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[2] Hagen Soltau, Hank Liao, and Hasim Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[3] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Interspeech*, 2017.

[4] Jan Chorowski and Navdeep Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *arXiv preprint arXiv:1612.02695*, 2016.

[5] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *International Conference on Machine learning (ICML)*, 2006, pp. 369–376.

[6] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning (ICML)*, 2014, pp. 1764–1772.

[7] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 167–174.

[8] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," *arXiv preprint arXiv:1412.1602*, 2014.

[9] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 577–585.

[10] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[11] Liang Lu, Xingxing Zhang, and Steve Renals, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5060–5064.

[12] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint CTC-attention based end-to-end speech recognition using multitask learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4835–4839.

[13] Takaaki Hori, Shinji Watanabe, and John R. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Association for Computational Linguistics (ACL)*, 2017.

[14] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahrmani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Interspeech*, 2016, pp. 2751–2755.

[15] Linguistic Data Consortium, "CSR-II (wsj1) complete," *Linguistic Data Consortium, Philadelphia*, vol. LDC94S13A, 1994.

[16] John Garofalo, David Graff, Doug Paul, and David Pallett, "CSR-I (wsj0) complete," *Linguistic Data Consortium, Philadelphia*, vol. LDC93S6A, 2007.

[17] Kikuo Maekawa, Hanae Koiso, Sadaoki Furui, and Hitoshi Isahara, "Spontaneous speech corpus of Japanese," in *International Conference on Language Resources and Evaluation (LREC)*, 2000, vol. 2, pp. 947–952.

[18] Yi Liu, Pascale Fung, Yongsheng Yang, Christopher Cieri, Shudong Huang, and David Graff, "HKUST/MTS: A very large scale Mandarin telephone speech corpus," in *Chinese Spoken Language Processing*, pp. 724–735. Springer, 2006.

[19] "VoxForge," http://www.voxforge.org/.

[20] Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Communication*, vol. 56, pp. 85–100, 2014.

[21] Tanja Schultz and Alex Waibel, "Fast bootstrapping of LVCSR systems with multilingual phoneme sets.," in *Eurospeech*, 1997.

[22] William Byrne, Peter Beyerlein, Juan M Huerta, Sanjeev Khudanpur, Bhaskara Marthi, John Morgan, Nino Peterek, Joe Picone, Dimitra Vergyri, and T Wang, "Towards language independent acoustic modeling," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2000, vol. 2, pp. II1029–II1032.

[23] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7304–7308.

[24] Georg Heigold, Vincent Vanhoucke, Alan Senior, Patrick Nguyen, M Ranzato, Matthieu Devin, and Jeffrey Dean, "Multilingual acoustic models using distributed deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8619–8623.

[25] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals, "Multilingual training of deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7319–7323.

[26] Zoltan Tuske, David Nolden, Ralf Schluter, and Hermann Ney, "Multilingual MRASTA features for low-resource keyword search and speech recognition systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7854–7858.

[27] Karel Veselỳ, Martin Karafiát, František Grézl, Miloš Janda, and Ekaterina Egorova, "The language-independent bottleneck features," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*, 2012, pp. 336–341.

[28] Javier Gonzalez-Dominguez, David Eustis, Ignacio Lopez-Moreno, Andrew Senior, Françoise Beaufays, and Pedro J Moreno, "A real-time end-to-end multilingual speech recognition architecture," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 749–759, 2015.

[29] Shigeki Matsuda, Xinhui Hu, Yoshinori Shiga, Hideki Kashioka, Chiori Hori, Keiji Yasuda, Hideo Okuma, Masao Uchiyama, Eiichiro Sumita, Hisashi Kawai, et al., "Multilingual speech-to-speech translation system: Voicetra," in *IEEE International Conference on Mobile Data Management (MDM)*, 2013, vol. 2, pp. 229–233.

[30] Ying Zhang, Mohammad Pezeshki, Philémon Brakel, Saizheng Zhang, Cesar Laurent Yoshua Bengio, and Aaron Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," *arXiv preprint arXiv:1701.02720*, 2017.

[31] Yu Zhang, William Chan, and Navdeep Jaitly, "Very deep convolutional networks for end-to-end speech recognition," *arXiv preprint arXiv:1610.03022*, 2016.

[32] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[33] Pegah Ghahremani, Bagher BabaAli, Daniel Povey, Korbinian Riedhammer, Jan Trmal, and Sanjeev Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 2494–2498.

[34] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in NIPS*, 2015.